Sl

# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
| --- | --- | --- | --- | --- |
| 09/729,508 | 12/04/2000 | Rene Vangemert | 00-487/1496.00053 | 9954 |

24319 7590 04/07/2004

LSI LOGIC CORPORATION
1621 BARBER LANE
MS: D-106 LEGAL
MILPITAS, CA 95035

| EXAMINER | |
| --- | --- |
| GERSTL, SHANE F | |
| **ART UNIT** | **PAPER NUMBER** |
| 2183 | 8 |

DATE MAILED: 04/07/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

SC

| Office Action Summary | Application No. | Applicant(s) |  |
|---|---|---|---|
|  | 09/729,508 | VANGEMERT ET AL. |  |
|  | Examiner | Art Unit |  |
|  | Shane F Gerstl | 2183 |  |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on 20 January 2004.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) 1-20 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-16,19 and 20 is/are rejected.

7)☒ Claim(s) 17 and 18 is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 20 January 2004 is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

## DETAILED ACTION

1. Claims 1-20 have been examined.

### *Papers Received*

2. Receipt is acknowledged of drawing and amendment papers submitted, where the papers have been placed of record in the file.

3. The amendment filed 20 January 2004 has successfully overcome the objections to the title, drawings and claims whereby the objections have been withdrawn.

4. The amendment filed 20 January 2004 has successfully overcome the 112 second paragraph rejections to claims 7 and 14 and have thus been withdrawn.

5. Applicant's arguments, see pages 11-12, filed 20 January 2004, with respect to the section headings have been fully considered and are persuasive. The objection of the specification has been withdrawn.

### *Claim Objections*

6. Claims 16-18 objected to because of the following informalities: the preamble states "The method of to claim 1" when it should read "The method of claim 1." Also, claim 16 indicates that multiple steps will be set forth due to use of the phrase "further comprising the steps of," but only one step is given in each claim.

Appropriate correction is required.

### *Claim Rejections - 35 USC § 102*

7. The rejections to claims 1, 3, 5, and 15 in regards to the merits are reproduced below with the same material arguments as presented in the previous Office Action.

8. Claims 1, 3, 5, 15, and 16 are rejected under 35 U.S.C. 102(b) as being anticipated by Mirapuri (5,590,294).

9. In regard to claim 1, Mirapuri discloses a method of recovering from invalid data in a first register within a pipelined processor (figure 5 shows pipelining), the method comprising the steps of:

a. setting a first status for said first register to an invalid state in response to an first data in said first register being invalid; In column 9, lines 35-42, Mirapuri discloses a set of control registers that tell whether a pipeline stage contains a valid or invalid instruction. Because the pipelined processor given by Mirapuri has eight stages and takes eight clock cycles to complete one instruction (column 5, lines 16-18), there are therefore at least eight pipeline registers. These registers hold the data for each stage that the control registers give the status of. Column 10, lines 56-58, show that data is loaded as invalid: "Often, the stall condition is only detected after parts of the pipeline have advanced using incorrect data."

b. stalling said processor in response to both (i) an instruction requiring said first data and (ii) said first status being in said invalid state. As shown above from column 10, lines 56-58, a stall condition is detected after invalid data has advanced, showing that the data must be needed since validity was not detected until now. Then after detecting the stall condition, Mirapuri stalls.

10. In regard to claim 3, Mirapuri discloses the method of claim 1, as described above, further comprising the step of: setting said first status to said invalid state in

response to receiving a second data from a second register having a second status in said invalid state. The case given above for claim 1 shows that if the pipeline has advanced invalid data, then whatever stage where the invalid data was detected has received its data from the previous pipeline stage and register.

11. In regard to claim 5, Mirapuri has disclosed the method of claim 4, as described above further comprising the step of: stalling said processor prior to obtaining said second data to prevent a third data from being written in said first register before said second data is written in said first register; As applied to claim 4, Mirapuri has disclosed a process that stalls if a register is invalid while obtaining valid data for it. Consequently, this valid data is obtained, and thus the pipeline stalled, before the data can be written to the register.

12. In regard to claim 15, Mirapuri discloses a pipelined processor (figure 5 shows pipelining) comprising:

a. means for buffering a first data; The pipelined processor given by Mirapuri has eight stages and takes eight clock cycles to complete one instruction (column 5, lines 16-18), there are therefore at least eight pipeline registers. These registers buffer data to the next stage of the pipeline.

b. means for buffering a first status; In column 9, lines 35-42, Mirapuri discloses a set of control registers that that tell whether a pipeline stage contains a valid or invalid instruction. The pipeline registers mentioned above contain the valid or invalid instruction. Thus, the control registers buffer the status of the pipeline registers.

c. means for setting said first status to an invalid state in response to said first data in said means for buffering being invalid; It is inherent that if the status registers of mention above hold validity data, then there is a means for setting the status. Column 10, lines 56-58, show that data is loaded as invalid when it says, "Often, the stall condition is only detected after parts of the pipeline have advanced using incorrect data."

d. means for stalling said processor in response to both (i) an instruction requiring said first data and (ii) said first status being in said invalid state. In column 6, lines 43-55, Mirapuri discloses the case of an instruction cache miss. Mirapuri states here that if there is a miss, an interlock occurs resulting in a stall. This miss detection is shown to not occur until after the fetch and decode stages (lines 49-52). Therefore, the pipeline registers of the fetch and decode stage are invalid when a miss occurs as Mirapuri shows, "Such stages are invalid at the time of the interlock," (column 10, lines 63-64).

13. In regard to claim 16, Mirapuri discloses the method of claim 1, further comprising the steps of: setting said first register status to said invalid state in response to a miss for a read from a data cache of a second data to be written in said first register. Column 5, lines 29-35 show that data is always fetched from a cache in the pipeline since the cache has dedicated stages for fetching from it. Therefore, if the first register is in one of the cache stages, data of the instruction is fetched from the data cache for storage in the pipeline registers so it can be processed. Column 6, lines 8-11 show that cache misses cause interlocks. As shown previously, an interlock is caused

by an invalid state of a pipeline stage and thus the cache miss sets the control register to invalid.

### *Claim Rejections - 35 USC § 103*

14. The rejections to claims 2, 7-10 and 12-14 in regards to the merits are reproduced below with the same material arguments as presented in the previous Office Action.

15. Claims 2 and 7 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mirapuri in view of Sites (5,193,167).

16. In regard to claim 2,

a. Mirupari discloses the method of claim 1, as described above, further comprising the step of: setting said first status to said invalid state (as described above).

b. Mirupari does not disclose the above function in response to a conditional store for said first data in said first register.

c. Sites has taught in column 3, lines 47-56 the use of a conditional store instruction. This instruction is shown to correspond to a matching locked load instruction. The store updates a memory position from the register loaded by the locked load instruction only if the register was not written to since the locked load instruction. In other words, the store conditional stores the contents of a register into memory based on the register's validity. This means that if the register was written to, the data in the register is not written to memory. Therefore, invalid data was loaded into the register. When one incorporates this disclosure of Sites

into the design of Mirapuri, one of ordinary skill in the art can see that when a conditional store is encountered and it is discovered that invalid data was loaded into the register, the status of the register would be set to invalid as shown above.

d. Sites has taught that this conditional store instruction permits the use of atomic byte writes (lines 47-49). This means that outside processes can only access the data in use before and after the transaction is complete. Otherwise, the store is not carried out. This data integrity that is provided during the execution of the conditional store instruction would have motivated one of ordinary skill in the art to implement the conditional store instruction taught by Sites in the design of Mirapuri.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Mirapuri to include the conditional store instruction taught by Sites in order to achieve data integrity upon executing a store.

17. In regard to claim 7, Mirapuri has disclosed the method of claim 1, as described above, further comprising the steps of:

a. setting said first status to said invalid state in response to at least one of (i) a conditional store for said first data in said first register, (ii) receiving a second data from a second register having a second status in said invalid state and (iii) a miss for a data cache read of a third data to be written in said first register.

i. Mirupari discloses the method of claim 1, as described above, further comprising the step of: setting said register status to said invalid state (as described above).

ii. Mirupari does not disclose that said register status for said register is set to said invalid state in response to a conditional store for said register.

iii. Sites has taught in column 3, lines 47-56 the use of a conditional store instruction. This instruction is shown to correspond to a matching locked load instruction. The store updates a memory position from the register loaded by the locked load instruction only if the register was not written to since the locked load instruction. In other words, the store conditional stores the contents of a register into memory based on the register's validity. This means that if the register was written to, the data in the register is not written to memory. Therefore, invalid data was loaded into the register. When one incorporates this disclosure of Sites into the design of Mirapuri, one of ordinary skill in the art can see that when a conditional store is encountered and it is discovered that invalid data was loaded into the register, the status of the register would be set to invalid as shown above.

iv. Sites has taught that this instruction pair permits the use of atomic byte writes (lines 47-49). This means that outside processes can only access the data in use before and after the transaction is complete.

Otherwise, the store is not carried out. This data integrity that is provided during the execution of the conditional store instruction would have motivated one of ordinary skill in the art to implement the conditional store instruction taught by Sites in the design of Mirapuri.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Mirapuri to include the conditional store instruction taught by Sites in order to achieve data integrity upon executing a store.

b. obtaining said third data from a memory in response to at least one of said conditional store and said miss; Since the previous section makes use of three limitations in the alternative and the first limitation has been met, there is no third data and this limitation is moot.

c. stalling said processor in response to obtaining said third data to enable said third data to be written in said first register; Since a previous section makes use of three limitations in the alternative and the first limitation has been met, there is no third data and this limitation is moot.

18. Claims 8, 10, 12-13, and 19-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mirapuri in view of Steiss (5,815,420).

19. In regard to claim 8,

a. Mirapuri discloses a pipelined processor (figure 5 shows pipelining), comprising:

i. A first register configured to buffer a first data

ii. logic configured to

set said first status for said register to an invalid state in response to said first data being invalid; In column 9, lines 35-42, Mirapuri discloses a set of control registers that that tell whether a pipeline stage contains a valid or invalid instruction. Because the pipelined processor given by Mirapuri has eight stages and takes eight clock cycles to complete one instruction (column 5, lines 16-18), there are therefore at least eight pipeline registers. These registers hold the data for each stage that the control registers give the status of. Column 10, lines 56-58, show that data is loaded as invalid: "Often, the stall condition is only detected after parts of the pipeline have advanced using incorrect data."

stall said processor in response to both (a) an instruction requiring said first data and (b) said first status being in said invalid state. As shown above from column 10, lines 56-58, a stall condition is detected after invalid data has advanced, showing that the data must be needed since validity was not detected until now. Then after detecting the stall condition, Mirapuri stalls.

b. Mirapuri does not disclose that the register is configured to buffer a first status. Mirapuri keeps track of register status in control registers that contain pipeline register validity information.

c. Steiss has disclosed a register configured to buffer a register status (figure 2, V bit). This status is used in the same manner as disclosed by Mirapuri for stalling the processor (Steiss, column 10, lines 9-11).

d. Having the register status bits buffered by the registers themselves allows for closer physical access of validity information to its corresponding data. This will then correspond to some speedup because of the physical closeness of needed information. This physical convenience and speedup would have motivated one of ordinary skill in the art at the time of invention to integrate the validity information of the registers in Mirapuri's disclosure into the register itself as described by Steiss.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Mirapuri to include the register status information in the register itself as given by Steiss for the purposes of convenience in location and logical speedup.

20. In regard to claim 10, Mirapuri in view of Steiss discloses the pipelined processor of claim 8, as described above, further comprising a second register, wherein said logic is further configured to set said first status to said invalid state in response to receiving a second data from said second register having a second status in said invalid state. The case given above for claim 8 shows that if the pipeline has advanced invalid data, then whatever stage where the invalid data was detected has received its data from the previous pipeline stage and register.

21. In regard to claim 12, Mirapuri in view of Steiss has disclosed the pipelined processor of claim 11, as described above, wherein said logic being further configured to

a. stall said processor while said bus interface unit is said obtaining said second data to prevent a third data from being written in said first register; As applied to claim 4, Mirapuri has disclosed a process that stalls if a register is invalid while obtaining valid data for it. Consequently, this valid data is obtained, and thus the pipeline stalled, before the data can be written to the register.

b. writing said third data said first register in response to writing said second data in said first register. When valid data from memory is written to the pipeline register of mention in Mirapuri indicating that the stall is completed, the pipeline continues its operation as normal and thus as the next instruction passes through each stage, new data is written to the register.

22. In regard to claim 13,

a. Mirapuri in view of Steiss, as applied to claim 8, discloses the pipelined processor of claim 8, as described above, that includes the register status in control registers that are checked on each clock cycle for indicating invalid states of pipeline stages;

b. Mirapuri in view of Steiss, as applied to claim 8, does not disclose that the register status is buffered as a plurality of bits to provide for multiple conditions that would set said register status to said invalid state.

c. Steiss has disclosed a pipelined processor (figure 1) that keeps track of register status. Steiss has disclosed that register status is buffered as a plurality of bits (figure 3, element 51) to provide for multiple conditions that would set said register status to said invalid state. In column 6, lines 50-56, Steiss shows the operation of the mentioned buffer as holding the status of multiple operands (conditions). In column 10, lines 9-11, Steiss has shown that these valid bits are used for stalling the processor, as is so in the disclosure of Mirapuri in view of Steiss as applied to claim 8.

d. Implementing the multiple bit validity buffer of Steiss in the design of Mirapuri in view of Steiss, as applied to claim 8, would allow each pipeline stage to realize invalid information for multiple conditions based on different operands. Upon realizing that the stage has received invalid data, the pipeline register would signify invalid data as described above. The ability to detect multiple reasons for invalidity would have motivated one of ordinary skill in the art to integrate the design of Steiss' validity buffer into that of Mirapuri in view of Steiss, as applied to claim 8.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Mirapuri in view of Steiss, as applied to claim 8, to include the validity buffer of Steiss in order to obtain the ability to detect multiple reasons for pipeline stage invalidity.

23. In regard to claim 19, Mirapuri in view of Steiss discloses the pipeline processor of claim 13, wherein said logic comprises a first logic gate configured to combine said

bits of said first status read from said first register. Column 12, lines 62-67 of Blomgren show that the valid bits are input into a logic network (and combined) to generate lengths (needed for address tracking).

24. In regard to claim 20, Mirapuri in view of Steiss discloses the pipeline processor of claim 8, wherein said logic comprises a first logic gate configured to combine a plurality of bits to form said first status written to said first register. Column 6, lines 26-29 show that the valid bit is set based on a comparison of values in an entry R. It is inherent that this comparison comprises a logic gate, and perhaps multiple gates, that combine the bits of the values being compared. This comparison serves as the basis for forming the first status written to the first register.

25. Claims 9 and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mirapuri in view of Steiss as applied to claim 8 above, and further in view of Sites.

26. In regard to claim 9,

a. Mirupari in view of Steiss discloses the pipelined processor of claim 8, as described above, wherein said logic being further configured to set said first status to said invalid state (as described above).

b. Mirupari in view of Steiss does not disclose the above function in response to a conditional store for said first data in said first register.

c. Sites has taught in column 3, lines 47-56 the use of a conditional store instruction. This instruction is shown to correspond to a matching locked load instruction. The store updates a memory position from the register loaded by the locked load instruction only if the register was not written to since the locked load

instruction. In other words, the store conditional stores the contents of a register into memory based on the register's validity. This means that if the register was written to, the data in the register is not written to memory. Therefore, invalid data was loaded into the register. When one incorporates this disclosure of Sites into the design of Mirapuri, one of ordinary skill in the art can see that when a conditional store is encountered and it is discovered that invalid data was loaded into the register, the status of the register would be set to invalid as shown above.

d. Sites has taught that this instruction pair permits the use of atomic byte writes (lines 47-49). This means that outside processes can only access the data in use before and after the transaction is complete. Otherwise, the store is not carried out. This data integrity that is provided during the execution of the conditional store instruction would have motivated one of ordinary skill in the art to implement the conditional store instruction pair taught by Sites in the design of Mirapuri in view of Steiss.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Mirapuri in view of Steiss to include the conditional store instruction pair taught by Sites in order to achieve data integrity upon executing a store.

27. In regard to claim 14, Mirapuri in view of Steiss has disclosed the pipelined processor of claim 8, as described above, further comprising:

a. A bus interface unit (figure 4, element 14) configured to obtain a third data for said first register from a memory; The memory management unit controls

accesses to and from the memory thus controlling the bus and acting as a bus interface unit. As described above, invalid data is at times loaded into the registers and when a register is invalid and accessed, the processor stalls. Mirapuri has shown in column 10, lines 63-65 that stages invalid at the interlock (stall) are preloaded with correct information. Because Mirapuri has shown in column 6, lines 54-56, one of multiple cases where correct data is loaded from memory, it has been shown that Mirapuri in view of Steiss loads correct information from memory when invalid data exists.

b. said logic is further configured to set said first status for said register to said invalid state in response to a conditional store for said first data in said first register.

i. Mirupari discloses the method of claim 1, as described above, further comprising the step of: setting said register status to said invalid state (as described above).

ii. Mirupari does not disclose the above function in response to a conditional store for said register.

iii. Sites has taught in column 3, lines 47-56 the use of a conditional store instruction. This instruction is shown to correspond to a matching locked load instruction. The store updates a memory position from the register loaded by the locked load instruction only if the register was not written to since the locked load instruction. In other words, the store conditional stores the contents of a register into memory based on the

register's validity. This means that if the register was written to, the data in the register is not written to memory. Therefore, invalid data was loaded into the register. When one incorporates this disclosure of Sites into the design of Mirapuri, one of ordinary skill in the art can see that when a conditional store is encountered and it is discovered that invalid data was loaded into the register, the status of the register would be set to invalid as shown above.

iv. Sites has taught that this instruction pair permits the use of atomic byte writes (lines 47-49). This means that outside processes can only access the data in use before and after the transaction is complete. Otherwise, the store is not carried out. This data integrity that is provided during the execution of the conditional store instruction would have motivated one of ordinary skill in the art to implement the conditional store instruction pair taught by Sites in the design of Mirapuri.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Mirapuri to include the conditional store instruction pair taught by Sites in order to achieve data integrity upon executing a store.

c. A second register; set said first status for said register to said invalid state in response to receiving a second data from said second register having a second status in said invalid state. The case given above for claim 1 shows that if the pipeline has advanced invalid data, then whatever stage where the invalid

data was detected has received its data from the previous pipeline stage and register.

d. stall said processor in response to obtaining said third data for said first register; In column 10, lines 59-63, Mirapuri has shown that parts of the pipeline are frozen (stalled) while others obtain corrected information. After receiving the corrected data, the pipeline registers of these stages are redone, but, the stalled stages continue to stall, thus in response to receiving valid data.

e. write said third data in said first in response to stalling said processor. As just described, when parts of the pipeline are stalled and others receive data, that data is then loaded to the pipeline registers.

f. set said first status to said invalid state in response to a cache load-miss for said first register. In column 6, lines 43-54, Mirapuri discloses the case of a cache miss. Mirapuri shows here that in such a case, the processor stalls. As shown above, the pipeline register is set to invalid when correct data is received by the bus interface unit.

28. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mirapuri in view of Blomgren (5,542,109).

29. In regard to claim 6,

a. Mirapuri discloses the method of claim 1 as shown previously.

b. Mirapuri does not disclose that the first status is buffered as a plurality of bits to provide for multiple conditions that would indicate said invalid state.

c. Blomgren has disclosed that the first status is buffered as a plurality of bits (figure 5; 1st, 2nd, 3rd valid entries) to provide for multiple conditions that would indicate said invalid state. Column 12, lines 23-35 show that the valid bits are encoded into first second and third fields each for representing an instruction. Table 2 in column 11 shows that each of the first two fields is composed of a plurality of bits that indicate multiple conditions. Column 11, lines 28-34 show that the valid fields are for indicating valid instructions (functional control words) in each stage of the pipeline. Column 3, lines 15-21 show that the valid bits indicate locations and order of valid instructions in the pipelines as a part of an address tracking means.

d. Column 1, line 65 – column 2, line 43 shows that address tracking is used for exception recovery and that the disclosed invention does so without storing each address in every stage, but, instead determines them whenever needed (using the valid bits shown above) and thus is low-cost. This ability to track instruction addresses at a low-cost would have motivated one of ordinary skill in the art to modify the design of Mirapuri to use the multiple valid field scheme disclosed by Blomgren. Referring to Table 2 again, one condition (00) would indicate invalid state when Blomgren is implemented into Mirapuri since there is no issued instruction in the stage at a certain time and thus the stage is invalid as shown previously. With Blomgren integrated into Mirapuri, the valid bits would be stored in the register as before and any other bits needed for address tracking may or may not be in the register.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Mirapuri to include the valid field scheme that includes a plurality of valid bits for each instruction as taught by Blomgren so that a low-cost address tracking scheme is realized.

### ***Response to Arguments***

30. Applicant's arguments, see page 6, filed 20 January 2004, with respect to the rejection(s)of claim(s) 6 under Mirapuri have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of Mirapuri in view of Blomgren.

31. Applicant's arguments filed 20 January 2004 with respect to the 35 USC 102 rejections of claims 1, 3, 8, and 15 have been fully considered but they are not persuasive.

32. In regard to the arguments for claims 1 and 15 on data stored in the registers, as noted by Applicant, Mirapuri does disclose in column 9, lines 35-42 that the pipeline registers of mention are instruction registers. All registers, including instruction registers hold or store data, in this case the data are instructions. The enclosed IEEE definitions of data include definitions where data is a quantity to which meaning has been assigned or a representation of instructions for processing by automatic means. In addition the Office has enclosed an IEEE definition of the term "register." The electronic computation definition shows that a register retains a portion of information (data or instructions in this case) of the entire system. Certainly instruction registers hold data in the form of instructions then. This means that the valid status defined in the Office

Action is applicable as the claimed invention stands. If Applicant meant for the claim to state that the registers hold operand data then it must be explicitly stated as such. Otherwise, Applicant is directed to the following paragraph.

33. Applicant is arguing a feature of the invention (data operand registers) not specifically stated in the claim language, which is improper. Claimed subject matter, **not** the specification, is the measure of invention. Limitations in the specification **cannot** be read into the claims for the purpose of avoiding the prior art. *In re Self*, 213 USPQ 1,5 (CCPA 1982); *In re Priest*, 199 USPQ 11,15 (CCPA 1978).

"It is the claims that measure the invention." *SRI Int'l v. Matshshita Elec. Corp.*, 775 F.2d 1107, 1121, 227 USPQ 577, 585 (Fed. Cir. 1985) (en banc).

"The invention disclosed in Hiniker's written description may be outstanding in its field, but the name of the game is the claim." *In re Hiniker Co.*, 47 USPQ2d 1523, 1529 (Fed. Cir. 1998).

"[A]s an initial matter, the PTO applies to the verbiage of the proposed claims the broadest reasonable meaning of the words in their ordinary usage as they would be understood by one of ordinary skill in the art, taking into account whatever enlightenment by way of definitions or otherwise that may be afforded by the written description contained in the applicant's specification." *In re Morris*, 44 USPQ2d 1023, 1027 (Fed. Cir. 1997).

"limitations appearing in the specification will not be read into the claims, and ... interpreting what is meant by a word in a claim 'is not to be **confused with adding an**

**extraneous limitation** appearing in the specification, which is improper'." *Intervet Am., v. Kee-Vet Labs.*, 12 USPQ2d 1474, 1476 (Fed. Cir. 1989)(citation omitted).

"it is entirely proper to use the specification to interpret what the patentee meant by a word or phrase in the claim, ... this is **not** to be confused with adding an extraneous limitation appearing in the specification, which is improper. By 'extraneous,' we mean a limitation read into a claim from the specification wholly apart from any need to interpret ... particular words or phrases in the claim." *In re Paulsen*, 31 USPQ2d 1671, 1674 (Fed. Cir. 1994) (citation omitted).

34. In regard to the arguments of claims 1, 8, and 15 on the Office Action appearing to be a reversing of cause and effect, please note the following. The cited section of column 10 shows that the stall condition is detected (and thus the pipeline stalls) only after invalid data has advanced through the pipeline. Further clarification of the correct cause and effect argued by the Applicant follows. Lines 63-64 show that the invalid stages are invalid at the time of interlock. Column 9, lines 20-34 show that the processor uses state machines and the control registers to define the pipeline state and lines 66-67 show that the control registers define the operation of the processor. Column 9, lines 35-42 show that the control registers specify valid or invalid instructions and thus valid or invalid pipeline stages. Since the state is defined by the control registers specifying valid or invalid data, moving into a stall state as specified in lines 20-34, column 10 lines 16-31, and figure 8 is in response to the control registers' valid or invalid specification. Since, the control registers specify the validity of the instructions in the pipeline registers of the pipeline and it is inherent that each pipeline passes the

this instruction data to the next pipeline. The data is then inherently needed or required by the next stage. Thus the stall condition occurs when the data is required and when the data is invalid.

35. In regard to the argument of claim 15 at the bottom of page 14, Applicant is again cautioned that the claims do not specify status bits in a register but merely that a means for buffering a status exists, which in this case is a control register that buffers a status (valid or invalid) as shown in column 9, lines 35-42. Please note paragraph 8 above. Since there is no claimed mention of status bits the Applicant's argument against the inherency is invalid. It is inherent that if the control registers hold a status that there exists a means for setting this status. If the claim mentioned status bits being set the argument would still not be persuasive. The control registers inherently contain bits. The status may then be called status bits. Whether the status bits are the whole register or not, they are set to valid or invalid as needed. And it would be inherent that the status bits are set.

36. In regard to the argument of claim 3 on pages 15-16, as shown above the control registers cause a stall when invalid data is detected. Column 10, lines 56-58 is cited to simply show that invalid data is advanced to other stages before detected at times. Since the data is instructions passed through the pipeline as shown above, data from a second register (preceding pipeline register) is at times sent to the register where when detected causes a stall as before. Since the stall is caused by the invalid data, the status must be set. This is further shown in column 9, lines 35-42 where the control

registers are shown to be a chain, thus propagating the status with the instructions from stage to stage.

37. Applicant's arguments filed 20 January 2004 with respect to the 35 USC 103 rejections of claims 2, and 7-10 have been fully considered but they are not persuasive.

38. In regard to the argument of claim 8 on pages 17-19, Mirapuri itself discloses (as shown above) that a register status (indicating valid or invalid) exists for pipeline registers (which hold data), just not that the status is inside the register as called for by the claims. Steiss shows in figures 2a and 2b that use of registers (R0....Rm) containing valid (V) bits within the registers. An obvious combination would have been to store the status (valid bits) inside the registers for which the validity is being indicated as taught by Steiss. As shown in the previous office action, one of ordinary skill in the art would have recognized that storing a status with the data in a register offers a locality convenience with respect to the data being accessed. In addition, to make integral (integrating the status discloses by Mirapuri into the register itself) is not patentable. MPEP 2144.04 (V-B) shows that In re Larson 144 USPQ 347 (CCPA 1965) set forth that combining two elements into one cannot be given patentable weight. Further, MPEP 2144.04 (VI-C) shows that In re Japikse 86 USPQ 70 (CCPA 1950) set forth that shifting the location of parts (shifting the valid status into the register of mention in this case) is not patentable. Therefore Mirupari in view of Steiss shows how the prior art is logically combined to get a register status stored in a register and one of ordinary skill in the art would recognize the motivation for combining the references, and the case law gives further indication of the obviousness.

39. In regard to the argument of claim 10 on pages 19-20, the argument given in the previous Office Action does meet the limitation of the claim. As shown previously, Mirapuri discloses in column 9 the use of chains of control registers, among which the second register lies. The second validity data of the second register is passed to the first register with the second data. Even if an invalid status from the second register is copied to the first register, this is in response to receiving the second data since the data and valid bits are received at the same time. Also, this "copying" function sets the first status of the first register to be the same as the second status of the second register and so the claim limitations are met.

40. In regard to the argument of claims 2, 7 and 9 on pages 20-21, the argument presented in the last Office Action cites the summary section of Sites. Perhaps the reference may be more completely understood by examining the detailed section given in column 14, line 64 – column 15, line 53. This section shows that the conditional store clears a lock flag once executed. The lock flag indicated that the store instruction was validly allowed to store data to a certain storage location. When the store completes, this lock or valid flag is set to zero indicating that a conditional instruction is not valid for storing to the location. In regard to the motivation, one must understand that data integrity is a degree of consistency or accuracy, as shown in the included IEEE standard definition of "data integrity," and not simply an entity that either exists or doesn't exist. Therefore, more data integrity is always desirable so that data is more and more consistent and accurate. Then it is shown in column 14, line 66 – column 15,

line 2 that the conditional store insures data integrity (gives a higher degree of data integrity).

### *Allowable Subject Matter*

41. Claim 4 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. The prior art of record does not specifically teach stalling said processor in response to *completing* said obtaining of data for a conditional store to enable said second data to be written in said first register. The prior art of reference fetches the data during the stall and thus cannot stall after the fetching is completed. In addition, no prior art of record suggests that it would have been obvious to one of ordinary skill in the art at the time of invention to modify the prior art of record to stall said processor in response to *completing* said obtaining to enable said second data to be written in said first register.

42. Claim 11 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. The prior art of record does not specifically teach stalling said processor in response to *completing* said obtaining of data to be written in said first register. The prior art of reference fetches the data during the stall and thus cannot stall after the fetching is completed. In addition, no prior art of record suggests that it would have been obvious to one of ordinary skill in the art at the time of invention to modify the prior art of record to stall said processor in response to *completing* said obtaining to enable said second data to be written in said first register.

43. Claim 17 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. The prior art of record does teach fetching data from a memory on a data cache miss as shown in column 7, lines 54-63. The prior art of record does not specifically teach stalling said processor in response to *completing* said obtaining to enable said second data to be written in said first register. The prior art of reference fetches the data during the stall and thus cannot stall after the fetching is completed. In addition, no prior art of record suggests that it would have been obvious to one of ordinary skill in the art at the time of invention to modify the prior art of record to stall said processor in response to *completing* said obtaining to enable said second data to be written in said first register.

44. Claim 18 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. All of the limitations of claim 17 are included in claim 18 and thus the claim is objected to on the same grounds as claim 17.

## *Conclusion*

45. The prior art made of record in the previous Office Action and not relied upon is considered pertinent to Applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane F Gerstl whose telephone number is (703)305-7305. The examiner can normally be reached on M-F 6:45-4:15 (First Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703)305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane F Gerstl
Examiner
Art Unit 2183

SFG
April 5, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100